Project 1: Image Filtering and Hybrid Images

Author: Gabriel Hofer

CSC-414 Introduction to Computer Vision

Instructor: Dr. Hoover

Due: February 26, 2020

Department: Computer Science and Engineering
University: South Dakota School of Mines and Technology

1 Questions

Q1: Explicitly describe image convolution: the input, the transformation, and the output. Why is it useful for computer vision?

Image convolution uses a kernel that is applied to each pixel in the image to create a new image. A kernel is used to view neighboring cells in the image and calculate a new pixel value for the resulting image.

Q2: What is the difference between convolution and correlation? Construct a scenario that produces a different output between both operations and show some images of the result. (you can use built in correlation functions and convolution functions in scipy here if desired).

Convolution takes two images and produces a new third image. Convolution is also a mathematical operation that takes to functions and creates a third function. Correlation measures the displacement of one signal relative to the other signal. It measures similarity of two images. Correlation is similar to convolution.

Q3: What is the difference between a high pass filter and a low pass filter in how they are constructed and what they do to the image? Please provide example kernels and output images.

The main difference between the two filters is the range of frequency that they allow to pass through them. A high pass filter passes signals with frequencies that are above a certain threshold. A low pass filter passes signals with frequencies that are below a certain threshold. Low pass filters can be created by taking an images and removing or subtracting the high-pass signals from it.

Q4: How does computation time vary with filter sizes from 1515 to 33 (for all odd and square sizes), and with image sizes from 0.25 MPix to 8 MPix (choose your own intervals)? Measure both using scipys convolve2d to produce a matrix of values (you should generate a 3d surface plot with image size as one axis, kernel size as the other axis, and time as the surface). You may use the skimage.rescale function to vary the size of an image.

As the kernel size increases, the amount of time required to perform the convolution also increases. This is because the algorithm to iterate through the kernel has complexity $O(N * M)$ where the kernel dimensions are N by M.